Computer Vision I

Project 2
October 17, 2017

Due on October 31, 2017

Image Mosaicing

The purpose of this project is to find corner features in multiple images and to align the images in a mosaic by estimating a homography between corresponding features.

1. **Overview:**

   In this project you will apply a Harris corner detector to find corners in two images, automatically find corresponding features, estimate a homography between the two images, and warp one image into the coordinate system of the second one to produce a mosaic containing the union of all pixels in the two images.

2. **Project Requirements:**

   (a) Write a program to implement the above ideas. Sample images to test your program will be available in blackboard. The basic outline of your program should be as follows:

      i. Read in two images. (Note: if the images are large, you may want to reduce their size to keep running time reasonable! Document in your report the scale factor you used.)

      ii. Apply Harris corner detector to both images: compute Harris R function over the image, and then do non-maimum suppression to get a **sparse set** of corner features.

      iii. Find correspondences between the two images: given two set of corners from the two images, compute normalized cross correlation (NCC) of image patches centered at each corner. (Note that this will be $O(n^2)$ process.) Choose potential corner matches by finding pair of corners (one from each image) such that they have the highest NCC value. You may also set a threshold to keep only matches that have a large NCC score.

      iv. Estimate the homography using the above correspondences. Note that these correspondences are likely to have many errors (outliers). That is ok: you should use RANSAC to robustly estimate the homography from the noisy correspondences:

         A. Repeatedly sample minimal number of points needed to estimate a homography (4 pts in this case).

         B. Compute a homography from these four points.

C. Map all points using the homagraphy and comparing distances between predicted and observed locations to determine the number of inliers.

D. At the end, compute a least-squares homgraphy from **ALL** the inliers in the largest set of inliers.

v. Warp one image onto the other one, blending overlapping pixels together to create a single image that shows the union of all pixels from both input images. You can choose which of the images to warp. The steps are as follows:

A. Determine how big to make the final output image so that it contains the union of all pixels in the two images.

B. Copy the image that does not have to be warped into the appropriate location in the output.

C. Warp the other image into the output image based on the estimated homography (or its inverse). You can use matlab functions if you want or write your own warping function.

D. Use any of the blending schemes we will discuss in class to blend pixels in the area of overlap between both images.

(b) **Write a report.** The report should include:

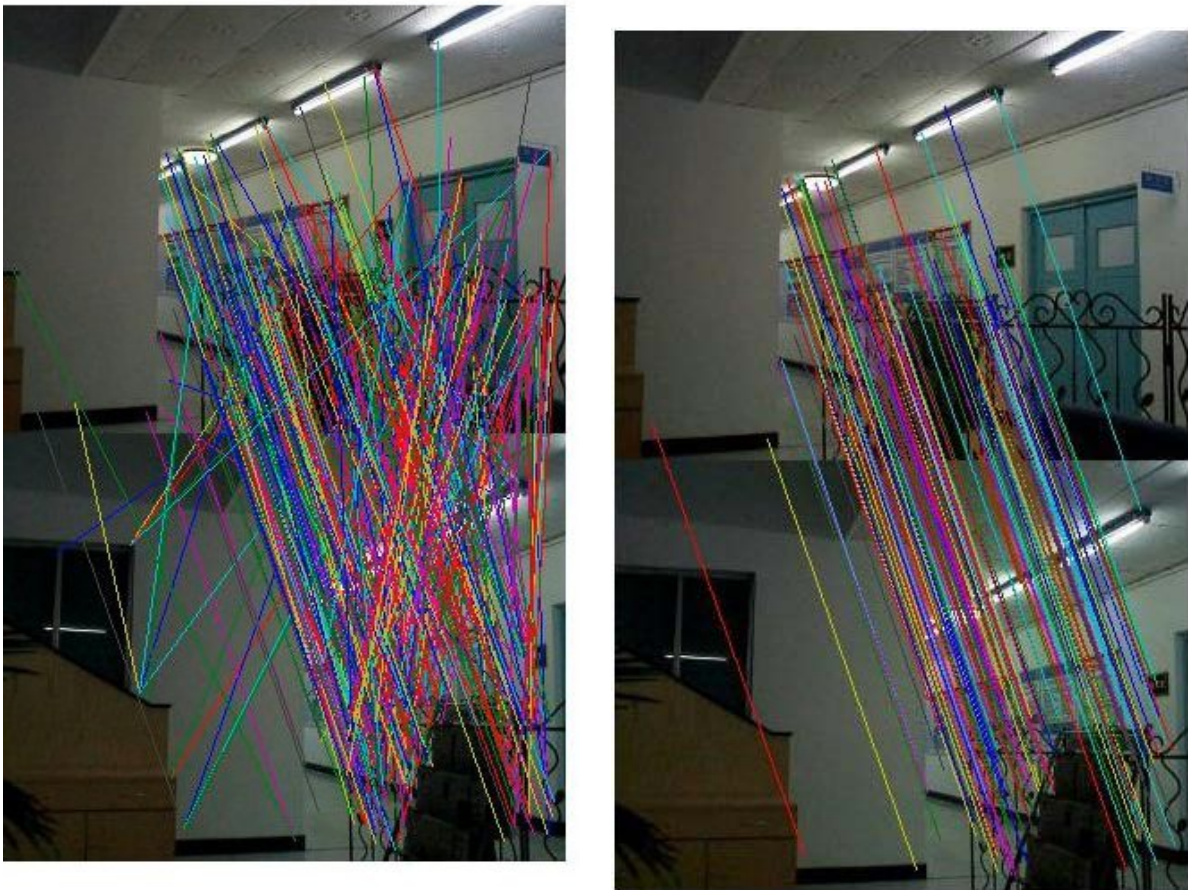
i. Abstract, description of algorithms, experiments, values of parameters used, observations and conclusions.

ii. A FLOWCHART, input and output images.

iii. An image showing potential corner feature location matches between the two images. An example of how to show them is given in the figure below.

iv. An appendix with your source code

Extra Credit Warp one image into a "frame" region in the second image. To do this, let the points from the one view be the corners of the image you want to insert in the frame and let the corresponding points in the second view be the clicked points of the frame (rectangle) into which the first image should be warped (Look at Matlab's `ginput` function for an easy way to collect mouse clicks). Use this idea to replace one surface in an image with a picture of your pet or project a drawing from one image onto the street in another image, or paste a powerpoint slide on a movie screen, etc ... Be creative! Send me your result by email so I can share it with the rest of the class.



(a) (b)

(c)

Figure 1: (a) Corresponding matches with outliers; (b) after they are cleaned and (c) the final mosaic.